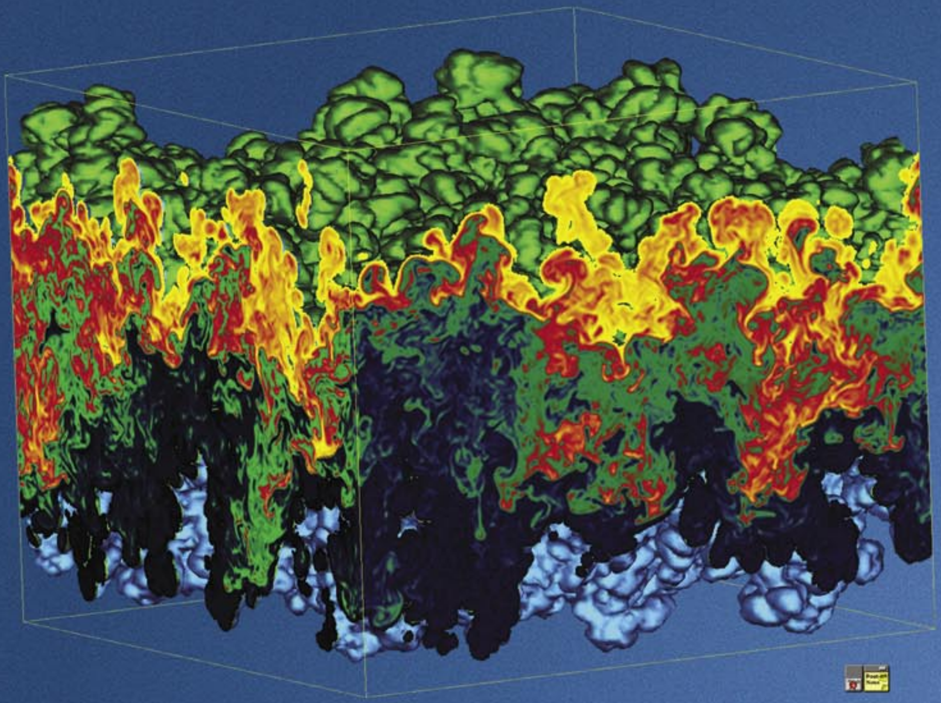
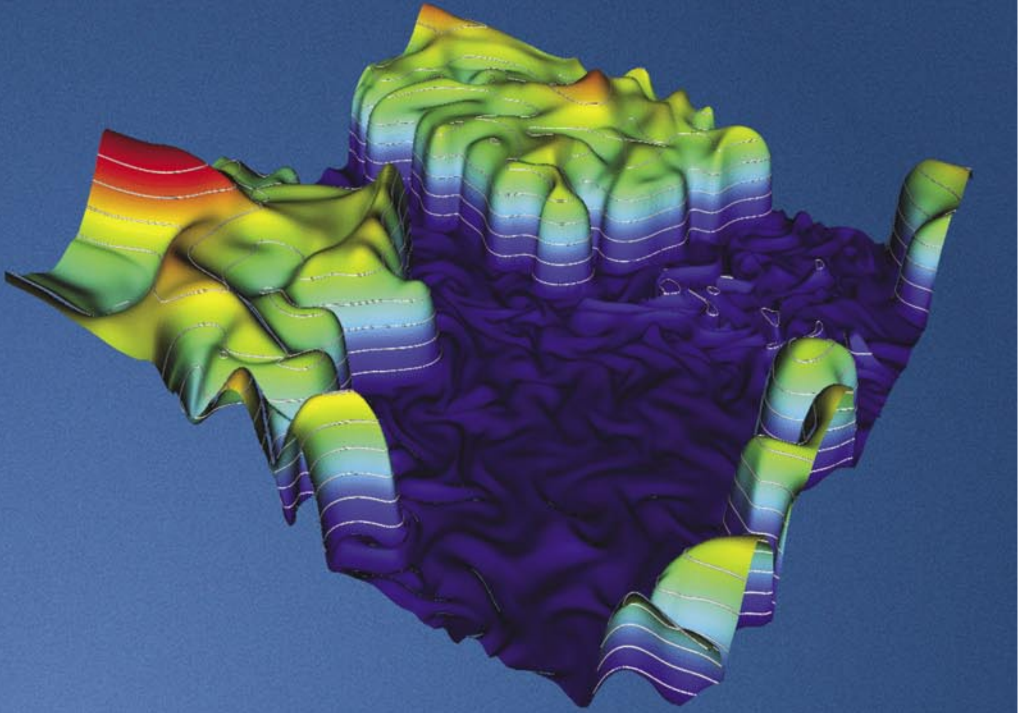
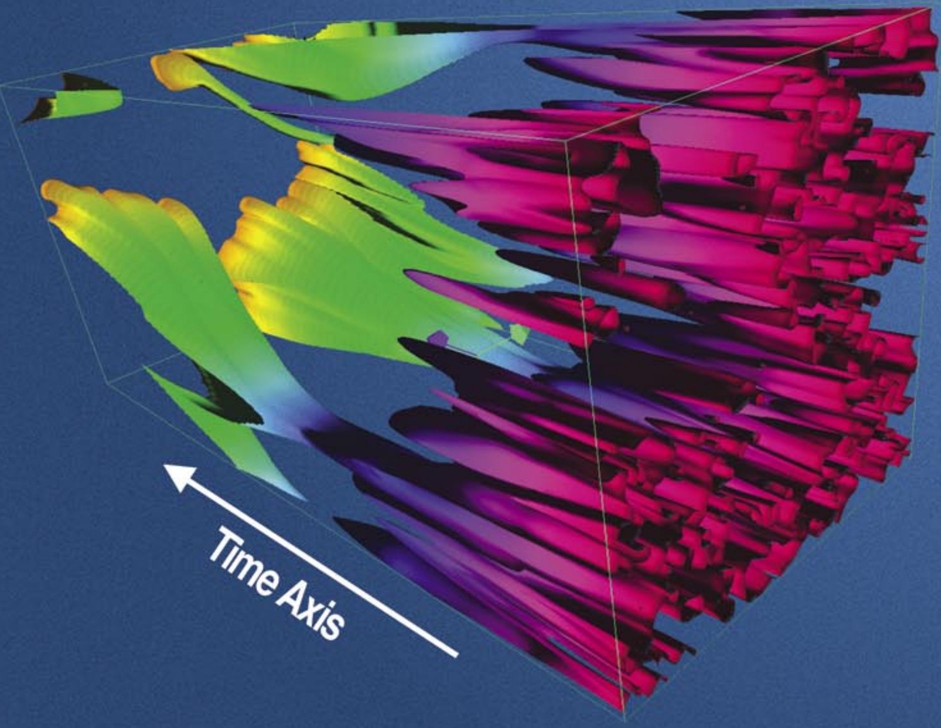
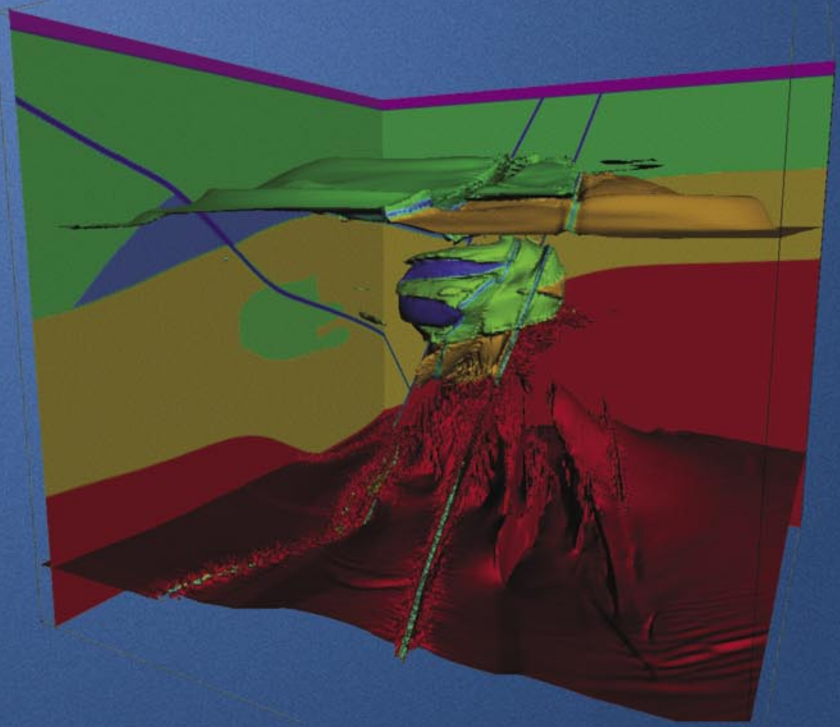
Section 4






Time Axis



4.00 Engaging in Computing Research and Advanced Development

The Computation Directorate's responsibility for enabling science goes beyond just helping solve today's problems; we must be prepared to solve tomorrow's known and anticipated problems through research and advanced development. We conduct collaborative scientific investigations that require the power of high performance computers and the efficiency of modern computational methods. Our research and development activities are applications-driven, and focused on LLNL programmatic objectives that require advanced computational technologies. This section highlights progress in a selection of projects from our portfolio of research and advanced development.

Much Computation Directorate Research and Development is characterized by the aggressive use of massively parallel computing to solve problems of national interest. The problems typically involve large-scale simulations of complex systems. Classic problems of scientific interest are usually based on Partial Differential Equations (PDEs) or continuum descriptions, but there is now increasing interest in systems that are better treated using discrete simulation. Additionally, many complex problems involve both multi-physics and multi-scale issues that demand a rethinking of not only the original formulation, but also the computer science design for codes that will efficiently use available resources. Figure 4.00-1 illustrates one such example. The first four reports in this Section describe recent progress addressing these types of issues.

Figure 4.00-1 (facing page). These simulations were created by the streaming process described in Section 4.10.

Performance continues to be another critical area of research. We are concerned with both our ability to exploit available computing platforms effectively and our ability to optimize the time required to write new codes, evolve codes over time to meet new mission demands, and move codes to new platforms as they become available. The goal of our research is to simplify the construction of re-useable software libraries and to improve the performance of existing scientific software. Current approaches include object-oriented design, scripting approaches for scientific simulations, and component technologies. The next three reports exemplify our work in this direction.

"Data Science" is our umbrella term for describing research over a wide range of topics related to understanding and effectively using large-scale data. The Laboratory is challenged to extract insight from massive amounts of data arising from numerous sources, including: scientific simulations, experimental devices, sophisticated sensor systems, specialized data bases, and public Web pages. Our goal is to enable scientists to concentrate on science by minimizing the burden of physically managing data and computer resources. This goal drives research efforts in terascale visualization, large-scale pattern recognition, clustering and classification algorithms, genetic and evolutionary algorithms, video and image analysis, feature extraction, query infrastructures and data access and integration in dynamic environments. The final three contributions in this Section illustrate our progress in Data Science.

To accomplish our research and advanced development objectives, the Computation Directorate partners extensively with academia and industry. We benefit significantly by engaging these groups in working with us to address our needs and objectives. Mechanisms for these partnerships take a variety of different forms. In 2003, Computation hosted 63 summer students, eight sabbatical visits, and 162 other visits by a total of 125 different visitors. We collaborate with universities and other national laboratories through jointly funded federal research proposals. We also fund some research subcontracts in direct support of on-going research projects within the Directorate. These interactions have contributed to the R&D results reported here and to the entire set of research activities in the Directorate.

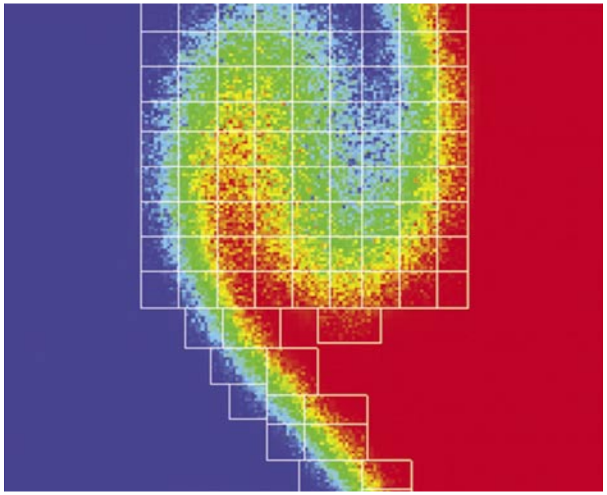


Figure 4.00-1. Hybrid multi-scale calculation using the SAMRAI (Adaptive Mesh Refinement) system. The fluid shear instability is resolved by employing continuum dynamics where applicable, and particle methods at the finest scale.

4.01 Scalable Linear Solvers

Problem Description

The B-Division code, ALE3D, uses implicit hydrodynamics techniques to generate structural dynamics simulations as part of the nation's Stockpile Stewardship Program. In 2003, the ALE3D team was interested in solving a very high-resolution spherical shell problem (Figure 4.01-1). The shell is composed of three layers and two different materials: steel for the inside and outside layers, and Lucite for the middle layer. One challenging aspect of this problem is the parallel solution of the linear systems that arise. In particular, the jumps in coefficients across material boundaries, the poor aspect ratios in the meshing of the steel layer, and the presence of so-called rigid body modes, create difficulties for the linear system solver.

Technical Approach/Status

The Scalable Linear Solvers (SLS) project is developing fast parallel multigrid algorithms and software for solving large, sparse linear systems of equations. The development of new linear solvers can often dramatically improve the capabilities of codes such as ALE3D, giving them the ability to simulate problems much faster and at much higher resolutions than ever before. Researchers are also investigating other numerical methods areas, including nonlinear solvers and sensitivity analysis. This research is contributing to programs at LLNL and elsewhere in the DOE, but for brevity, we have focused on one specific highlight in this report.

Progress in 2003

During 2003, two solver advances helped the ALE3D team accomplish record-breaking simulations. The first was the development of an automatic scheme for choosing smoothing parameters in algebraic multigrid. Using multigrid convergence theory, we derived formulas for pseudo-optimal smoothing parameters. These formulas require estimates for the largest eigenvalue of the smoother, which are computed by employing a known relationship between the conjugate gradient method and the Lanczos eigenvalue method. The second advance was the development of a new solver based on the smoothed-aggregation method. This latter solver exploits the availability of the rigid body modes and requires less memory and computations per iteration than the algebraic multigrid solver.

Significance

These solver advances have so far enabled solution of the spherical shell problem in Figure 4.01-1 on meshes with more than half a billion (610 million) degrees of freedom on 4032 processors of ASCI White in less than half an hour. This is 100 times larger, and run on 10 times the number of processors, than the simulations of only three years ago. It is also the largest implicit hydrodynamics calculation done to date in the ALE3D code.

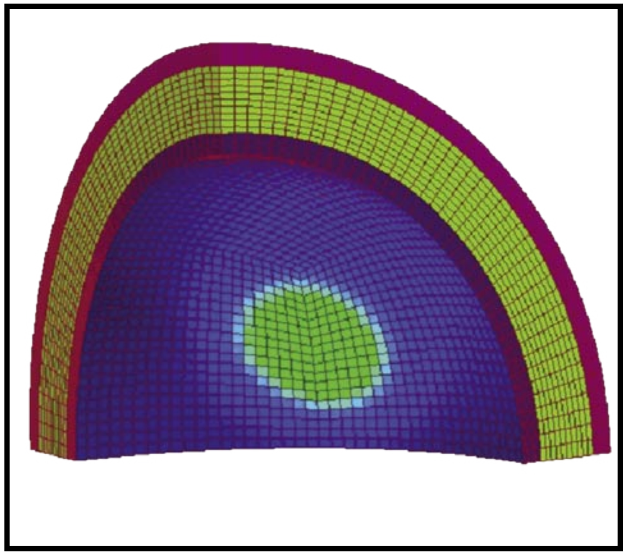


Figure 4.01-1. This simulation of a spherical shell implicit hydro problem used 4 million elements and 12.3 million unknowns. Advances in 2003 allow up to 610 million unknowns.

Problem Description

Many applications require the development and analysis of numerical methods for high-fidelity PDE-based simulations involving complex, possibly moving geometry. New techniques that address this application regime must be flexible, computationally efficient, and highly accurate.

Technical Approach/Status

We are developing advanced grid generation and discretization techniques that deliver highly flexible geometry representations while retaining the accuracy and efficiency advantages of simple single-block structured grids. In particular, we develop methods for 1) adaptive overlapping grids which consist of structured, logically rectangular curvilinear body-fitted component grids that overlap where they meet; 2) mixed-element or hybrid grids where the grid consists primarily of large regions of logically rectangular structured mesh; and 3) embedded boundary grids that represent complex geometry by cutting a structured component grid with a complex surface. In each case, we use highly efficient discretization techniques in structured grid regions and develop new techniques to handle grid overlap, mixed element meshes, or general polyhedral cells.

Progress in 2003

For overlapping grids, we developed new algorithms and grid generation capabilities for solving the incompressible Navier–Stokes equations for airflows around stadiums and cityscapes (Figure 4.02-1). We also performed basic research that led to a new understanding of stability properties of incompressible Navier–Stokes solutions. To develop overset grid techniques for free boundary dynamics, we analyzed non-Newtonian viscous fingering and developed novel time-stepping algorithms for coupling an elastic boundary to an incompressible fluid. With the Chemistry & Materials Science Directorate, we developed a 3D multi-block solver for rapid model prototyping of biochemical reactions using Overture software.

For embedded boundary methods, we developed a second-order accurate method for the second-order wave equation in general 2D domains. For the Neumann problem, we analyzed stability using a normal mode technique; for the Dirichlet problem, we developed a discrete boundary stencil that avoids the small-cell time-step restriction and devised the smooth startup procedure necessary to obtain second-order accurate gradients. We combined these techniques to solve Maxwell's equations written as a system of second-order wave equations for general 2D domains.

For hybrid grids, we designed and implemented 2D/3D unstructured , mixed element, second-order finite volume mesh operators, integrated them into Overture, and verified their accuracy. Using these operators, we completed preliminary work to solve Maxwell's equations on 2D mixed element meshes.

Significance

This work enables the solution of incompressible flow applications of dispersive modeling problems important to DHS, as well as shedding new light in basic research areas in biological computing and turbulence modeling. New hybrid and embedded boundary methods are expected to be of critical importance to maintain accuracy and efficiency as geometric domains become more complicated in applications such as accelerator modeling.

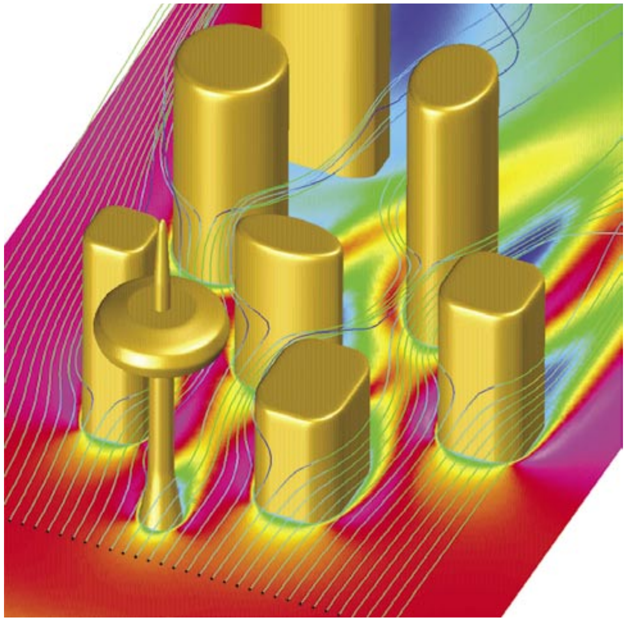


Figure 4.02-1 This flexible geometry simulation models dispersion in a cityscape domain.

4.03 SAMRAI

Problem Description

The SAMRAI project focuses on enabling structured adaptive mesh refinement (SAMR) technology for large-scale parallel applications. Simulations of science and engineering problems often require high resolution on multiple scales, both spatial and temporal. SAMR systematically focuses computational resources by dynamically adjusting resolution of the computational mesh.

Technical Approach/Status

The SAMRAI project is a research base for application, numerical software, and parallel computing issues associated with SAMR. The SAMRAI software library is employed in application development in various projects at LLNL and other institutions. SAMRAI software simplifies parallel SAMR development for computational scientists and allows software technology to be leveraged across multiple applications. Its object-oriented design allows SAMRAI capabilities to be enhanced and extended to meet unique problem requirements.

Project collaborations at LLNL involve problems in fluid dynamics, electronic structures, and adaptive ALE hydrodynamics. University collaborations involve hybrid continuum–particle methods, industrial fire simulations, and computational biology.

Progress in 2003

The ALE–AMR code couples SAMR and ALE methods using SAMRAI (Figure 4.03-1). The DNT Directorate is increasingly interested in this work. Initially, we are integrating SAMRAI capabilities into the DNT CALE code; we hope that this effort leads to adoption of SAMRAI and adaptive methods in other DNT codes. The DHS is using SAMRAI to develop high-resolution simulation models for releases of airborne contaminants in urban settings. This project involves the FEM3MP dispersion code from ASD, and complex geometry facilities of the Computation Directorate Rapsodi project.

Another project uses SAMRAI to develop adaptive methods for electronic structures calculations that will scale much better than existing state-of-the-art approaches. Our algorithm research crosscuts several projects. Recently developed combinatorial techniques significantly reduce adaptive meshing and communication costs in large-scale parallel applications. Nearly optimal scaling has been shown for canonical adaptive computations using more than 1000 processors (Figure 4.03-2).

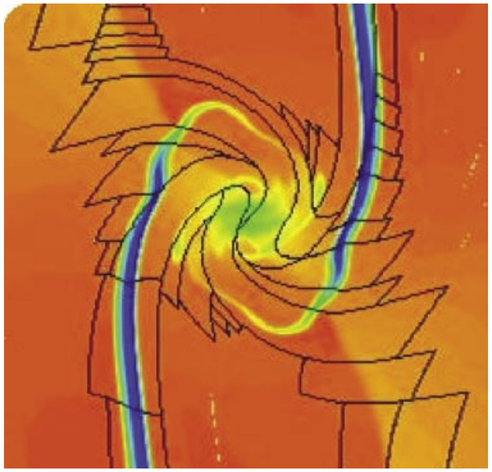


Figure 4.03-1. ALE–AMR combines deforming meshes and mesh refinement.

In 2003, we also organized an Institute for Scientific Computing Research (ISCR) workshop on multiscale methods, gathering researchers from DOE Labs, academia, and industry to discuss multiscale simulation techniques and their applicability to LLNL programs.

Significance

New and growing collaborations demonstrate that AMR is important for large-scale computational problems, and that SAMRAI is a viable software platform for programmatically-driven application development. By focusing computational resources, high-resolution simulations are achieved more efficiently than static mesh approaches that often yield inadequate resolution.

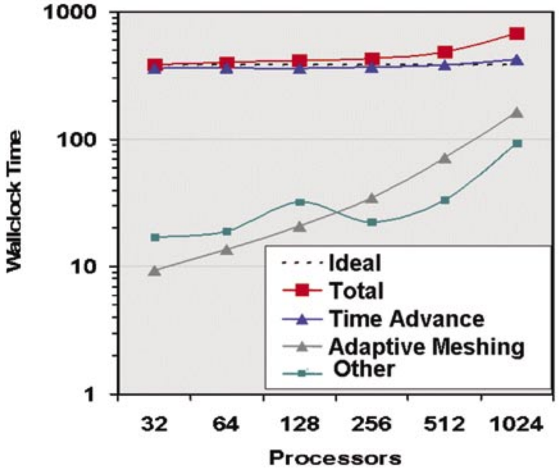


Figure 4.03-2. SAMRAI adaptive meshing operations show NlogN scaling on more than 1000 processors.

4.04 Tiling Models for Spatial Decomposition in AMTRAN

Problem Description

The AMTRAN code project works to scale deterministic (S_n) neutron transport problems up efficiently to thousands of parallel processors by means of spatial domain decomposition and load balancing for a finite-element code with block-structured adaptive mesh refinement on Cartesian grids. This project is of particular interest to DNT.

Technical Approach/Status

A directed binary spatial decomposition scheme was found that can be coupled with predetermined scripts for assigning subdomains to processors and for orchestrating the order of computational steps as these subdomains are repeatedly "swept" sequentially from multiple directions in parallel. These scripts can be represented diagrammatically as (logically) space-filling tiles composed of rectangles, each representing a computational unit of one sweep direction on a subdomain on a single sub-iteration (Figure 4.04-1). Identically shaped tiles are fitted together, like the pieces of a jigsaw puzzle, to minimize idle time as much as possible. As the degree of spatial parallelism and the number of processors increase, the tiles grow increasingly complex in a fractal-like manner.

Progress in 2003

The tiling method was discovered and implemented in 2003 by generalizing some simpler known cases of optimal scheduling and by assigning multiple subdomains to individual processors. The algorithm has been tested on cases involving from 4 subdomains in 2 dimensions to 512 subdomains in 3 dimensions, and scripts up to 4096 subdomains are ready for future parallel architectures. A simple example is shown in Figure 4.04-1 for a 4-by-4 set of domains in 2 dimensions, where efficiency is doubled from 40% to 80% by overlaying two half-tiles on one another. A plot of efficiencies of various configurations in three dimensions is reproduced as Figure 4.04-2. It shows that efficiencies approach 100% asymptotically as we successively double the number of subdomains for a given degree of parallelism. The three curves illustrate (from top to bottom) 16-way, 32-way, and 64-way spatial parallelism, respectively. The term "maximum theoretical processor usage" in the title means that we assume perfect load balance and ignore communication overhead costs.

Significance

The tiling method has led to significant improvements in efficiency and parallel scalability for the AMTRAN code. In addition, it provides for the first time a theoretical basis for projecting performance on more massively parallel computers planned for the future.

Master	Subdomain	Sub-iteration										Master	Subdomains	Sub-iteration									
(without subdomain overloading)												(with subdomain overloading)											
		1	2	3	4	5	6	7	8	9	10			1	2	3	4	5	6	7	8	9	10
		Sweep Directions												Sweep Directions									
1	(1,1)	0			1			2			3	1	(1,1), (3,1)	0		2	1	0	3	2	1		3
2	(1,2)		0	1					2	3		2	(1,2), (3,2)		0	1	2	3	0	1	2	3	
3	(1,3)		1	0					3	2		3	(1,3), (3,3)		1	0	3	2	1	0	3	2	
4	(1,4)	1			0			3			2	4	(1,4), (3,4)	1		3	0	1	2	3	0		2
5	(2,1)			0		2	1		3			5	(2,1), (4,1)	2		0	3	2	1	0	3		1
6	(2,2)				0	1	2	3				6	(2,2), (4,2)		2	3	0	1	2	3	0	1	
7	(2,3)				1	0	3	2				7	(2,3), (4,3)		3	2	1	0	3	2	1	0	
8	(2,4)			1		3	0		2			8	(2,4), (4,4)	3		1	2	3	0	1	2		0
9	(3,1)			2		0	3		1														
10	(3,2)				2	3	0	1															
11	(3,3)				3	2	1	0															
12	(3,4)			3		1	2		0														
13	(4,1)	2			3			0			1												
14	(4,2)		2	3					0	1													
15	(4,3)		3	2					1	0													
16	(4,4)	3			2			1			0												

Figure 4.04-1. A 4-by-4 set of domains in two dimensions, where efficiency is doubled from 40% to 80% by overlaying two half-tiles on one another

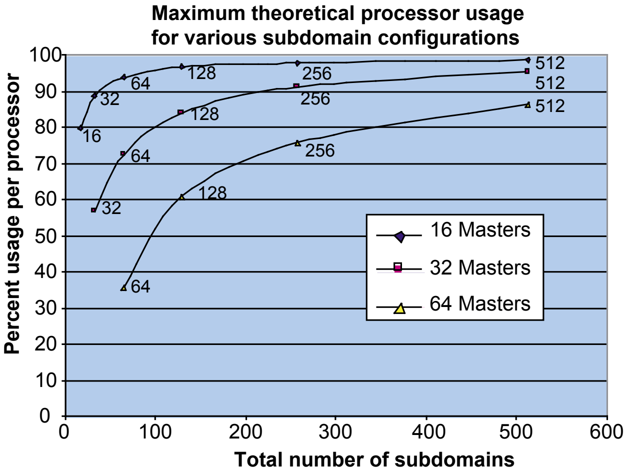


Figure 4.04-2. Theoretical efficiencies of various configurations in three dimensions.

4.05 Performance of Multi-Physics Applications

Problem Description

Supercomputer performance is often quoted in terms of the Linpack benchmark, which carries out dense linear algebra computations. By this measure, some systems can achieve 60–90% of their peak theoretical performance. ASCI applications, by contrast, typically use multiple coupled physics algorithms, each of which places unique demands on the computer. These applications are considered excellent performers when they reach 15% of peak on a single CPU for a given algorithm, or 5–8% aggregate over a time step. Understanding and explaining this dichotomy between benchmark and“real” application performance was a high priority in 2003, as several external review panels asked LLNL to study this topic.

Technical Approach/Status

Our approach to this problem was two-fold. First, we took detailed measurements of the algorithmic characteristics of their codes. This was done through a collaboration of code developers from DNT and computer scientists from the PSE/Tools Group. This group, known as APOMP (ASCI Performance Optimization and Modeling Project), was chartered to study and improve application performance. Characteristics such as CPU instruction mix, computational intensity (flops per memory op), cache hit ratios, parallel efficiency, and more were collected using a host of tools.

Second, a very simple model was developed to help determine why attaining performance near peak levels on a microprocessor-based architecture was impossible without significantly changing the algorithmic characteristics of the physics.

Progress in 2003

This model was applied first to the Power3 microprocessor architecture, the foundation of ASCI White. Peak performance on that chip requires two FMA (fused floating-point multiply-add) instructions to be issued every cycle, for a peak speed of 4 flops/cycle. This in turn requires the program to use each operand it fetches from memory three times for every result it stores.

By carefully measuring the instruction mix and computational intensity of the ASCI applications, we were able to show an upper bound on single-CPU performance of 23% of peak for data culled from several actual ASCI milepost calculations. This model did not take into account cache misses, integer instructions, parallel efficiency, or a host of other potential factors that would further close the gap between predicted and actual peak performance.

Significance

This simple model was presented to several external review committees (the JASONs, the ASCI Burn Code Review Committee, and the National Research Council), all of whom were keenly interested in application performance. By showing that we are reaching performance close to the“empirically derived upper bound” of the chip (versus theoretical peak performance), we allayed the concerns of the external reviewers and laid the groundwork to define new metrics for understanding the performance of ASCI applications.

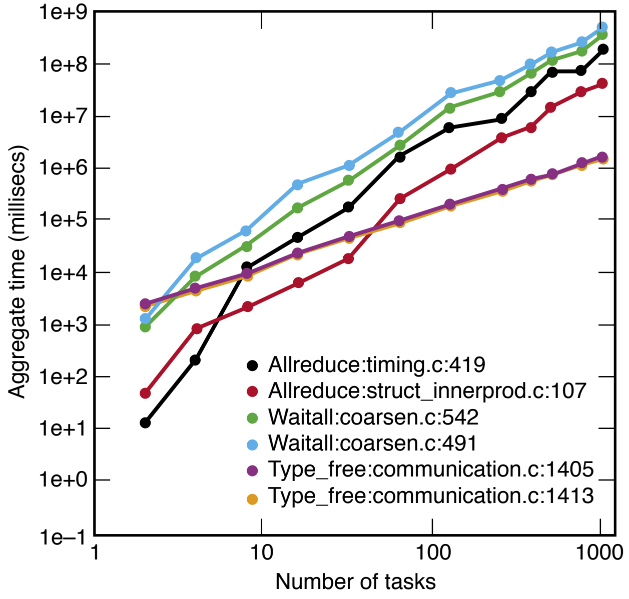


Figure 4.05-1. The mpiP tool shows how much time various MPI calls take as the number of tasks increases. Steep increases indicate possible scaling problems.

Problem Description

A major LLNL strategy in satisfying the need for production-quality capacity high-performance computing (HPC) cycles is the deployment of large-scale Linux clusters. These clusters take advantage of low-cost commodity processor components, which provides significantly more computing capacity than traditional proprietary HPC solutions. In conjunction with and critical to the success of this strategy is the development and deployment of an open source, production-quality Linux software stack.

Technical Approach/Status

The LLNL Linux software stack consists of three major components: an operating system, a parallel file system, and a resource management system. CHAOS is an in-house Red Hat-based Linux distribution that includes modifications for high-performance networks, cluster management and monitoring, and access control. Lustre is an open source high-performance parallel file system developed in part through a collaboration between LLNL and Cluster File Systems, Inc. (CFS) directed at achieving high performance within a single computing cluster, as well as in a shared environment. SLURM is a tool developed by LLNL and Linux NetworX to manage a queue of pending work, allocate access to nodes, and launch and manage parallel jobs.

Progress in 2003

CHAOS 1.2 was released in 2003 and is installed on all LLNL production Linux clusters. Several new or enhanced CHAOS components were released including a scalable, intra-cluster authentication service *(munge)* that enables secure and confidential application-level communication; a secure remote shell based on munge (*mrsh*); a multicast-based cluster heartbeat service *(whatsup)*; a cluster configuration database (*genders*); and enhancements to the power management software (powerman). In addition, a pre-release of the 64-bit CHAOS 2.0 distribution needed for Thunder includes improved Itanium processor machine check support.

Lustre was first deployed in production on MCR including, late in the year, a multi-net capability that allows the PVC cluster to directly share MCR's file system, the first shared parallel file system at LLNL and an important step in enabling inter-system high-performance file sharing. Lustre was also deployed on a portion of the classified network. The reliability and performance of Lustre were improved significantly through the efforts of CFS and intensive testing by LLNL on the ALC cluster.

SLURM was installed on every Linux cluster at LLNL including MCR, ALC, PVC, Lilac, iLX, and Ace. It has proven to be reliable and highly scalable, with excellent performance characteristics. As a result of its success on Linux-based systems, SLURM is now being ported to AIX to deploy on the ASCI platforms in lieu of the existing resource management software.

Significance

LLNL's Linux software stack is critical to deploying production-quality HPC resources, and 2003 proved to be a major turning point in providing a production-capable software stack. Major advances made in CHAOS, Lustre, and SLURM have made production-quality, large-scale HPC Linux clusters a reality.

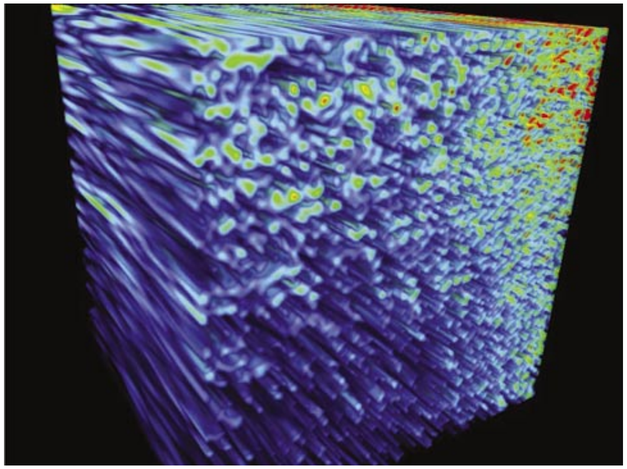


Figure 4.06-1. PF3D Simulation: 1920 processors on MCR cluster performed February 2003 (1/16th cross-section shown here).

4.07 Advancing the Code Development Environment

Problem Description

The goal for the parallel code development environment is to deliver the software tools and runtime libraries that allow applications to execute efficiently on the high-end computing platforms at LLNL.

Technical Approach/Status

Our parallel code development environment exploits a combination of local R&D and partnership development with academic and commercial sources. We test external products and provide requirements to external developers. The new large Linux clusters and planning for BG/L and Purple systems dominate our efforts. The LLNL user community insists that we stretch the limits of size and scalability, at the same time improving functionality and efficiency.

Progress in 2003

BG/L, a massively parallel cellular architecture system being developed jointly by IBM and LLNL, represents a significant architectural change from current ASCI systems. Adaptation of LLNL applications to this hardware prior to its delivery to LLNL in FY05 is critical to the overall project success. We are facilitating that adaptation through the use of BGLsim, a system simulator for parallel machines developed by IBM for hardware validation and software development.

Figure 4.07-1 shows that BGLsim models the complete BG/L hardware and system software environment. As a result, porting and tuning of applications on BGLsim directly correspond to results that will be seen on the as-yet-unbuilt machine. LLNL contributed to BGLsim by modifying the software so that it can use a variety of MPI implementations. In particular, LLNL modifications support the use of the native Quadrics MPI on Élan-based Linux systems, including the ASCI Linux Cluster. The simulator is installed on LLNL systems, and is instrumental in porting LLNL and ASCI Alliance applications to prototype BG/L hardware installed at IBM's Yorktown Heights facility.

The ROSE compiler project focused on sophisticated or domain-specific source-to-source translators that optimize existing scientific applications in C and C++. ROSE can trigger the generation of low-level platform-specific code to provide high performance, while preserving the simplicity and productivity of high-level of abstractions for the developer. Accomplishments during 2003 included the six-fold speedup of a C++ application, numerous loop optimizations, program analysis, and documentation.

The Etnus Totalview partnership added memory debugging functionality and a major speedup in launch of large jobs among the new product features. The Tool Gear infrastructure project created an interactive graphical interface for the mpiP communication-profiling tool to users' view source code annotated with communication performance information on both Linux and IBM AIX systems. Joint work with IBM on MPI and OS scalability work continued with development and test of a new kernel for the AIX OS that co-schedules interfering OS activities that are limiting scalability of collective operations

Significance

The thrust of effort in this area pushes the limits of scale, preparing for use of larger systems and also support of the large ASCI codes.

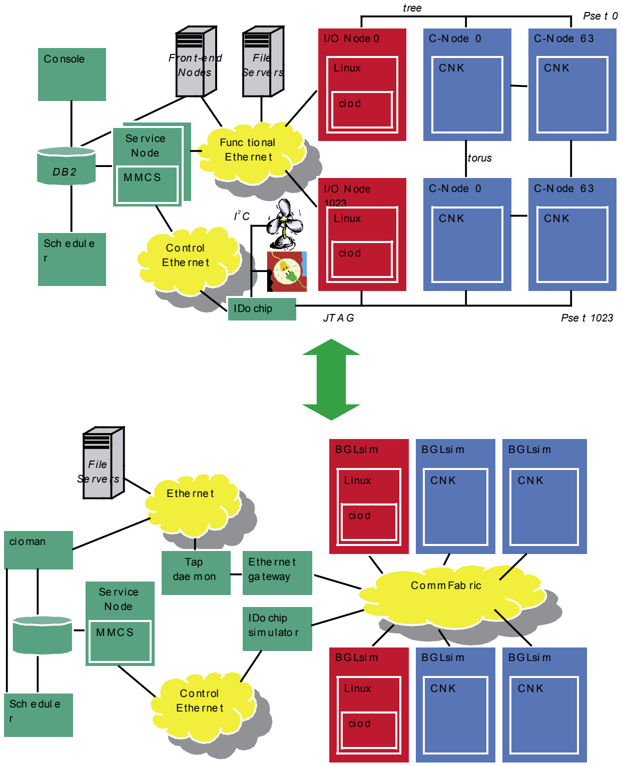


Figure 4.07-1. BGLsim, a system simulator for parallel machines developed by IBM for hardware validation and software development, models the complete BG/L hardware and system software environment.

Problem Description

A significant challenge in scientific data management is to improve scientists' interactions with huge data sets. Scientist's desire for ever-increasing resolution of their simulations is supported by ongoing increases in computational power. As a result, current simulations commonly produce data sets well over100 GB, and the average data set size is increasing. As data sets grow, it becomes increasingly difficult for scientists to gain insight from the data, because there is too much information to understand. To address this problem, scientists need a way to focus on relevant data and to eliminate unimportant information.

Technical Approach

To intelligently filter simulation data, we are developing an approximate, *ad hoc* query infrastructure. This infrastructure provides range-based queries against simulation data to identify regions of interest and produces a new data set containing only those regions. If an optional preprocessing step is performed, query accuracy can be traded for time, with the best answer possible within the given time constraints being returned.

Progress in 2003

A complex topology-based agglomeration algorithm was designed and implemented. A multi-resolution view of a data set is generated by repeatedly applying this algorithm to the corresponding mesh topology. An incremental model creation algorithm was also designed and implemented, so that statistical models of the data could be created at the same time the hierarchy was built.

A single-pass clustering algorithm was designed and implemented. This algorithm identifies regions (i.e. nodes in the topology-based hierarchy) that are similar in multi-dimensional space and groups them together.

A beta version of the *ad hoc* query infrastructure that queries mesh files directly is currently in limited deployment. While it does not support trading time for accuracy, it is a parallel implementation that easily queries data sets of several hundred gigabytes. It has performed queries on 200-GB data sets such as shock wave tracking and selection of regions performing complex chemistry calculations.

Significance

The hierarchy creation and model-generation algorithms allow us to create multi-resolution hierarchies and models for complex mesh types, including unstructured and adaptive meshes. This dramatically increases the generality of our prototype.

The clustering algorithm will be used to improve the overall query performance by collocating similar regions on disk, so fewer reads are required to return relevant regions.

By performing a limited deployment of our prototype, we are able to make a direct impact on scientists' ability to understand their data, while also obtaining valuable insight into additional desired capabilities. This insight will allow us to focus our future research on the most valuable activities.

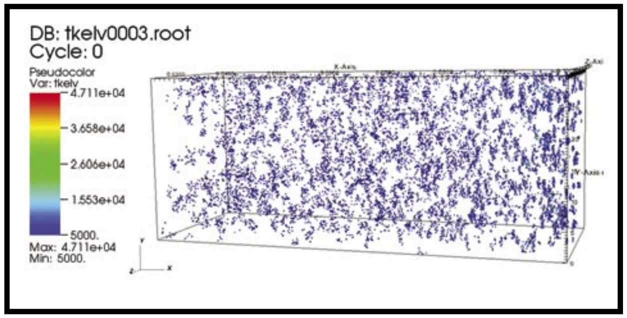


Figure 4.08-1. Regions within a time step in which complex chemistry calculations were performed.

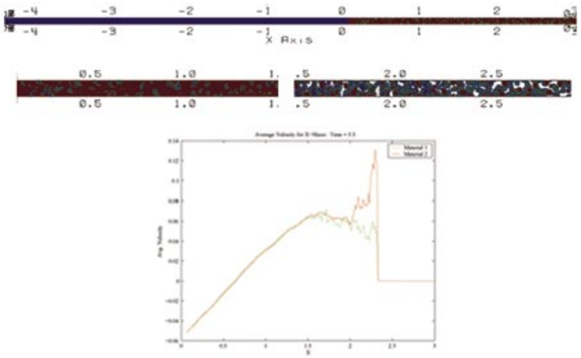


Figure 4.08-2. The top mesh is queried to generate the two meshes in the middle, and these meshes are analyzed to yield the resulting plot.

4.09 Scalable Scientific Data Mining

Problem Description

Advances in technology permit scientists to gather data from experiments, simulations, and observations at an ever-increasing pace. These massive, complex data sets are available as time-series or as images. Since it is impractical to manually analyze, explore, and understand this data, useful information is often overlooked, and the potential benefits of increased computational and data gathering capabilities are only partially realized.

Technical Approach/Status

The Sapphire project addresses the challenge of data overload by applying and extending ideas from the multi-disciplinary field of data mining. We conduct research in algorithms, incorporate the research into software, and apply the software to real-world problems at LLNL. The needs of these applications, in turn, drive our research.

We define data mining as the end-to-end process of extracting useful information from raw data. We focus on the compute-intensive activities—processing the data to extract objects and relevant features, dimension reduction techniques to identify key features, and pattern recognition techniques—to identify and characterize patterns in the data, which are then shown to the scientist for validation.

Our focus has evolved from the development of algorithms and software, to include the application of the software to problems of interest. Our object-oriented software currently supports all steps in the data mining process, with several algorithms provided for each step. The software is serial; many parts are embarrassingly parallel, and additional support for parallelism will be provided as required by the applications.

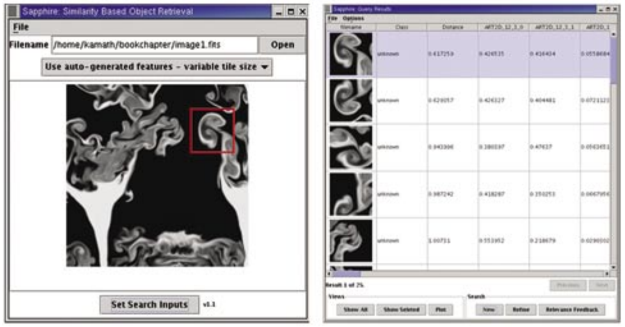
Progress in 2003

Our primary effort was in applications. We developed a Similarity-Based Object Retrieval (SBOR) system for retrieving objects in image and mesh data that are similar to a given query object (Figure 4.09-1). We are collaborating with AX Division on a code validation problem, and we are working with physicists at the DIII-D Tokamak to identify key features associated with the quiescent H-mode in the plasma. We began work on the detection and tracking of moving objects in video, a technique of interest in surveillance and computer simulations. We are also collaborating on computer security problems with CIAC.

We also investigated several algorithms for feature selection, the use of texture features for more efficient retrieval of high-resolution, remote-sensing imagery, and improved techniques for background subtraction to detect moving objects (Figure 4.09- 2).

Significance

Our progress in 2003 has helped LLNL scientists in several directorates, enabling us to apply cutting-edge data analysis techniques to their problems. Several of these problems are rather difficult and require further development of innovative approaches. Our research and involvement in professional activities continue to place Sapphire at the forefront of the scalable scientific data-mining field.



Filename /home/kamath/bookchapter/image1.fits

Use auto-generated features – variable tile size

Figure 4.09-1. The Sapphire Similarity-Based Object Retrieval (SBOR) system allows a user to identify an object of interest in an image (left) and returns similar objects in the image database, ranked by similarity (right). This example shows that the system is invariant to rotation and translation.

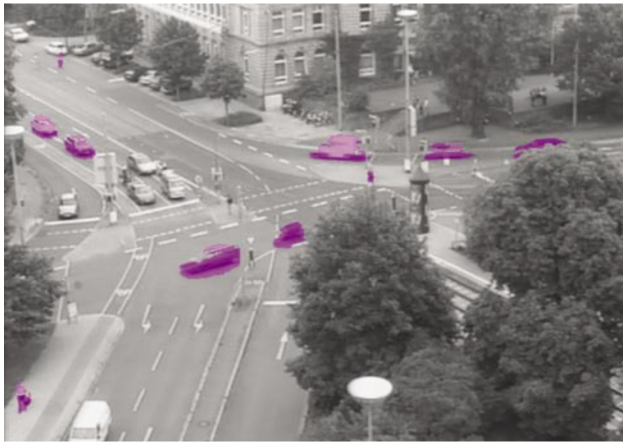


Figure 4.09-2. A frame from a video, with the moving objects highlighted. These objects were detected using background subtraction algorithms.

4.10 Scalable Interactive Data Exploration Tools

Problem Description

Laboratory scientists create very high-resolution models at an ever-increasing rate, which translates into their need to explore terabytes of data in real time. The challenge is to produce interactively any view of a high-resolution model by accessing the minimum amount of data. Moreover, one must help the user understand reliably the fundamental structures present in the model by exploring the minimum number of views.

Technical Approach

In response to this trend, the visualization group is implementing a long-term research plan based on the combination of two main strategies: redefinition of the visualization pipeline as a streaming process based on progressive and cache oblivious algorithms, and development of new data analysis tools that are tightly coupled with the visualization process and guide the user in navigating the data. This new visualization pipeline allows developing software tools that are intrinsically scalable with the size of the input problem and the performance of the computing resources. In our complementary research strategy, we introduce data analysis tools computing intrinsic topological and metric properties that help the user understand the structure of the data.

Progress in 2003

We completed our first prototype of the ViSUS Progressive Viewer, demonstrating the unprecedented capability of effectively exploring large data sets (e.g., 8-billion-node mesh) with resources as modest as a laptop computer or as large as a parallel visualization server driving a PowerWall display. We generated the Figure 4.10-1 images on a Dell laptop. Furthermore, we connected our streaming infrastructure to simulation codes developed independently with minimal code intrusion (one function call at the end of the time steps visualized).

Based on a Morse–Theoretical framework, we introduced algorithms for efficient and stable computation of critical points in a scalar field, their organization in the Morse–Smale Complex, and their persistence at different levels of resolution. The work is moving toward the definition of data comparison metrics and the analysis of dynamic structures.

Significance

Our streaming infrastructure optimizes the use of human and computing resources by providing three major new capabilities: large-scale visualization on low-end computers, remote visualization, and real-time monitoring of parallel simulations. A scientist can explore data sets that are tens to hundreds of GB on an office desktop workstation. Using a 10-MB network, we can access in real time remote data sets that are terabytes in size. We reduced our preprocessing time to a negligible amount so that even for high-resolution simulations we can have one time-step ready for visualization as the next one is still being computed. This enables new capabilities such as early detection and the elimination of wasteful executions.

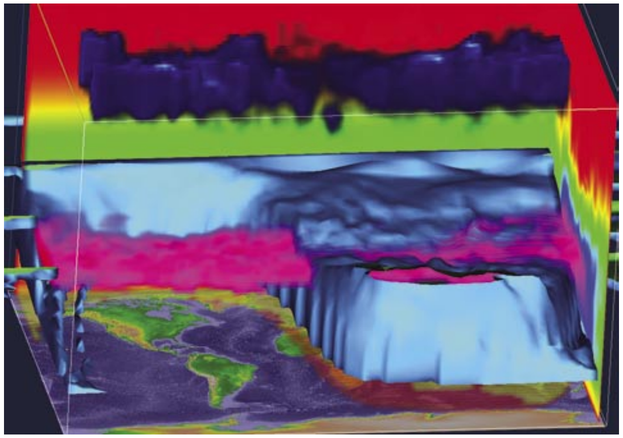


Figure 4.10-1. Researchers can explore terabytes of data in real time, via a streaming-infrastructure visualization pipeline.

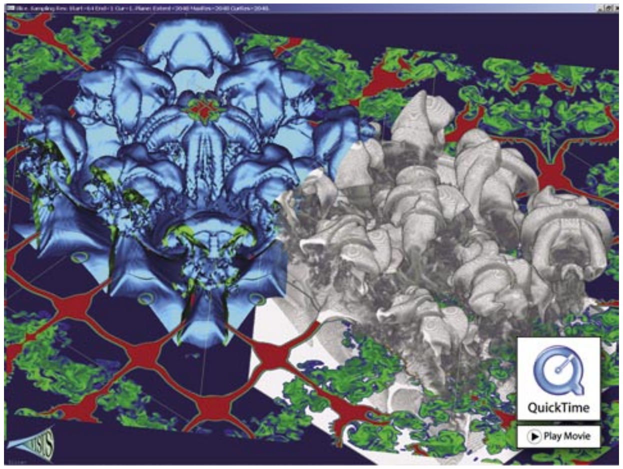


Figure 4-10-2. PPM Simulation, winner of a Gordon Bell Prize.